

WHAT IS CLAIMED IS

1. A method of transferring data between a processing agent and a memory resource comprising:

designating the memory resource for pushing the data to the processing agent using a push bus having a plurality of sources that arbitrate use of the push bus; and,

designating the memory resource for receiving the data from the processing agent using a pull bus having a plurality of destinations that arbitrate use of the pull bus.

2. The method of claim 1 wherein transferring comprises: establishing a plurality of contexts on the programming agent and maintaining program counters and context relative registers.

3. The method of claim 2 wherein the programming agent executes a context and issues a read command to a memory controller in a read phase.

4. The method of claim 3 wherein the memory controller processes the read command to be sent to the memory resource.

5. The method of claim 4 wherein the context is swapped

out if the read data is required to continue the execution of the context.

6. The method of claim 5 wherein after the memory controller has completed the processing of the read command, the memory controller pushes the data to an input transfer register of the programming agent.

7. The method of claim 6 wherein after the data has been pushed, the programming agent reads the data in the input transfer register and the programming agent continues the execution of the context.

8. The method of claim 2 wherein the programming agent executes a context and loads the data into an output transfer register of the programming agent in a write phase.

9. The method of claim 8 wherein the programming agent issues a write command to a memory controller and the output transfer register is set to a read-only state.

10. The method of claim 9 wherein the context is swapped out if the write command is required to continue the execution of

the context.

11. The method of claim 10 wherein the memory controller pushes the data from the output transfer register and the memory controller sends a signal to the programming agent to unlock the output transfer register.

12. The method of claim 11 wherein if the context has been swapped out after the output transfer register has been unlocked, the context is swapped back in and the programming agent continues the execution of the context.

13. A system comprising:

a memory resource;

a processing agent configured to transfer data between the processing agent and the memory resource;

a push bus for pushing the data to the processing agent having a plurality of sources that arbitrate use of the push bus;

a pull bus for receiving the data from the processing agent having a plurality of destinations that arbitrate use of the pull bus; and

a plurality of microengines executing multiple contexts that seek resources of the processing agent.

14. The system of claim 13 further comprising a read phase for the transfer of data from the memory resource to the processing agent in which the transfer is unidirectional.

15. The system of claim 13 further comprising a write phase for the transfer of data from the processing agent to the memory resource in which the transfer is unidirectional.

16. The system of claim 13 further comprising a plurality of program counters and a plurality of context relative registers.

17. The system of claim 16 in which the context relative registers are selected from a group comprising of general purpose registers, inter-programming agent registers, static random access memory (SRAM) input transfer registers, dynamic random access memory (DRAM) input transfer registers, SRAM output transfer registers, DRAM output transfer registers, and local memory registers.

18. The system of claim 17 in which the programming agent is configured to execute a context and issue a read command to a

memory controller.

19. The system of claim 18 in which the memory controller is configured to process the read command to be sent to the memory resource.

20. The system of claim 19 in which the programming agent is configured to swap the context out if the read command is required to continue to execution of the context.

21. The method of claim 20 in which after the read command is processed, the memory controller is configured to push the data to an input transfer register of the programming agent and the programming agent is configured to read the data in the input transfer register and to continue the execution of the context.

22. The system of claim 15 in which the programming agent is configured to execute a context and load the data into an output transfer register of the programming agent.

23. The system of claim 22 in which the programming agent is configured to issue a write command to a memory controller and in which the output transfer register is set to a read-only

state.

24. The system of claim 23 in which the programming agent is configured to swap the context out if the write command is required to continue to execution of the context.

25. The system of claim 24 in which the memory controller is configured to push the data from the output transfer register and to send a signal to the programming agent to unlock the output transfer register.

26. A computer program product residing on a computer readable medium for causing a parallel processor to perform a function comprises instructions causing the processor to:

designate the memory resource for pushing the data to the processing agent using a push bus having a plurality of sources that arbitrate use of the push bus; and,

designate the memory resource for receiving the data from the processing agent using a pull bus having a plurality of destinations that arbitrate use of the pull bus.

27. The computer program product of claim 26 further comprising instructions causing the processor to establish a

plurality of contexts on the programming agent and maintaining program counters and context relative registers.

28. The computer program product of claim 26 wherein the programming agent in a read phase executes a context and issues a read command to a memory controller.

29. The computer program product of claim 26 wherein the memory controller processes the read command to be sent to the memory resource and the context is swapped out if the read command is required to continue the execution of the context.